

# **EXHIBIT 3**



# A Survey of Encryption Standards

Numerous encryption standards dot the microcomputer landscape, seemingly covering every application. One nevertheless finds much common ground underlying the many standards. This survey discusses the standards and their algorithms, how they compare, how they differ, and where they're headed.

Burt Kaliski

RSA Laboratories

**C**ryptography is the science, or some would say the art, of secret codes. In its broadest sense cryptography addresses a number of practical problems:

- *confidentiality*, keeping messages secret;
- *origin authentication*, verifying a message's source;
- *integrity*, assuring that a message has not been modified; and
- *key management*, distributing the secret "keys" for cryptographic algorithms.

This survey focuses on encryption algorithms, the low-level, step-by-step transformations on messages that address these problems, as well as applications that involve encryption. It covers both approved standards and work in progress; the modifiers *draft* and *proposed* should help with the distinction.

Since descriptions here are at a summary level, readers seeking greater depth may refer to the standards documents or to encryption surveys such as those by Diffie,<sup>1</sup> Simmons,<sup>2</sup> which includes a reprint of Diffie's article, and Fahn,<sup>3</sup> which is available from RSA Laboratories or via anonymous ftp to rsa.com. Patel gives an earlier survey on security standards for the Open Systems Interconnection (OSI) reference model.<sup>4</sup>

Much of the encryption standards work fits into one or more security "models." The models do

not specify algorithms; rather, they define services and give structures for encryption protocols. The OSI Security Architecture standard<sup>5</sup> is one helpful reference. Also on the road to international standardization is the Generic Upper Layers Security (GULS) standard.<sup>6</sup> GULS forms the basis for IEEE P802.10, a local-area network security project, and the draft ANSI X9.41,<sup>7</sup> a standards effort for electronic data interchange.

Many ways other than encryption exist to protect data, from access control to tamper-resistant coatings, but they are outside the scope of this article. Even in systems based on cryptography, other issues than just the codes come into play, such as random number sources and password selection guidelines. The US Department of Defense's "Orange Book" is one of many helpful references for these topics.<sup>8</sup>

Remember, draft standards and other works in progress are subject to change. Furthermore, with the large number of standards efforts, I may not have covered some relevant efforts. An effort's absence from this article in no way minimizes its importance.

## Algorithms

An encryption algorithm is a method of transforming a message to add some cryptographic protection, such as confidentiality or integrity. Most encryption algorithms involve one or more keys, which are cryptographic variables, often

unique to one user, that control the algorithm and provide security against attackers.

Cryptographers often classify encryption algorithms according to the type of transformation and keys. Each class solves a different set of cryptographic problems. Some classes require that parties first agree on a secret key by secure means that are separate from the normal communication protocol; others do not have this limitation. I describe the algorithms standards according to one such classification: secret-key cryptosystems, public-key cryptosystems, digital signature schemes, key-agreement algorithms, cryptographic hash functions, and authentication codes. Table 1 summarizes the classes and their properties.

**Secret-key cryptosystems.** These algorithms encrypt and decrypt messages with a key in such a way that it is difficult to decrypt without the key. Because the encryption and decryption keys in a secret-key cryptosystem are the same, such systems are often called symmetric in the literature.

Most secret-key cryptosystems operate on messages one block at a time; a block may be 64 bits long, and the keys are usually short, say, 56 bits long. Ideally, an attacker's only approach is trial and error, which amounts, for example, to  $2^{56}$  trials for 56-bit keys. Secret-key algorithms are generally quite fast.

Secret-key cryptosystems provide confidentiality and key management to parties who have previously agreed on a secret key. The Data Encryption Standard (DES)<sup>9</sup> is the primary standard. Published in 1977 and recently affirmed for a fourth five-year period, DES defines the Data Encryption Algorithm (DEA). It also specifies how to implement DEA: in hardware. Technically, software implementations of DEA, which abound, do not comply. ANSI standard X3.92<sup>10</sup> and Australian Standard AS2805.5<sup>11</sup> specify DEA.

Despite much controversy about the nature of DEA—the government never revealed its design criteria—the algorithm seems to be quite secure, as far as 56-bit algorithms go. It resists powerful attacks that have broken other systems.<sup>12,13</sup>

Along with DES come some standard modes of operation, including electronic codebook, cipher block chaining, cipher feedback, and output feedback.<sup>14</sup> These modes apply to any block cipher, not just DEA. ANSI X9.17<sup>15</sup> introduces the encrypt-decrypt-encrypt (EDE) mode of encryption involving two DEA keys.

Two password-based encryption algorithms defined in the intervender public-key cryptography standard (PKCS) #5<sup>16</sup> are also based on DEA.

A potential new standard secret-key cryptosystem is Skip-

**Table 1. Encryption algorithm classes and their properties.**

Class	C	OA	I	KM	Prior
Secret-key cryptosystems	Yes	No	No	Yes	Yes
Public-key cryptosystems	Yes	No	No	Yes	No
Digital signature schemes	No	Yes	Yes	No	No
Key-agreement algorithms	Yes	Optional	No	Yes	No
Cryptographic hash functions	No	No	Yes	No	No
Authentication codes	No	Yes	Yes	No	Yes

C indicates confidentiality; OA, origin authentication; I, integrity; KM, key management.  
Prior requires that parties first agree on a secret key.

jack, a classified part of the proposed escrowed encryption standard.<sup>17</sup> A panel of cryptography experts recently certified Skipjack, with 80-bit keys, as appearing secure,<sup>18</sup> but its details remain unpublished.

Secret-key cryptosystems are rarely standardized; some standards bodies explicitly omit them from their scope. One of the few other candidates is RC4, a fast secret-key cryptosystem with variable-length keys.<sup>19</sup> RC4 is adopted in the cellular digital packet data (CDPD) specifications.<sup>20</sup>

**Public-key cryptosystems.** These algorithms encrypt and decrypt messages with two different keys in such a way that it is difficult to decrypt without the decryption key. The encryption key can be published without compromising security, and is called the public key for this reason; the decryption key is called the private key. Because the encryption and decryption keys in a public-key cryptosystem differ, such systems are often called asymmetric in the literature. The idea comes from Diffie and Hellman.<sup>21</sup>

Public-key cryptosystems provide confidentiality and key management. They can be as secure or more secure than secret-key cryptosystems, but they are generally slower. Their main advantage is that, since the encryption key can be published, parties need not first agree on a secret key. They are often combined with secret-key cryptosystems to gain the benefits of both: speed without prior secrets.

Although there is no primary standard public-key cryptosystem, many consider a cryptosystem invented by Rivest, Shamir, and Adleman (RSA)<sup>22</sup> in 1977 a de facto standard. Public-key cryptosystems, like secret-key cryptosystems, are rarely standardized; when they are standardized, key management is a more likely purpose than confidentiality.

Efforts toward RSA standardization include the intervender PKCS #1,<sup>23</sup> which gives block formats for RSA operations, and the draft ANSI X9.31 part 4,<sup>24</sup> which is currently based on PKCS #1. PKCS #1's block formats have been adopted by Internet privacy-enhanced mail<sup>25</sup> and, among other algorithms,

Encryption standards**Glossary**

The acronyms for encryption standards and the groups developing them are considered by some as a form of encryption in its own right. Following is an abridged "key" to the various acronyms and their meanings, as well as to several standards organizations.

<b>ASC X9</b>	Accredited Standards Committee X9 (Financial Services), a body that develops standards for the banking industry; accredited by ANSI	<b>GULS</b>	Generic Upper Layers Security, an OSI security architecture effort
<b>ANSI</b>	American National Standards Institute, an organization that accredits standards bodies	<b>IEC</b>	International Electrotechnical Commission, an international standards body
<b>CCITT</b>	Comité Consultatif International de Télégraphique et Téléphonique, (International Telegraph and Telephone Consultative Committee), an international standards body	<b>IEEE</b>	Institute of Electrical and Electronics Engineers, an organization that develops transnational standards; that is, the standards are the consensus of individuals rather than national representatives
<b>CFONB</b>	Comité Français d'Organisation et de Normalisation Bancaire, a French banking standards body	<b>Internet</b>	A transnational body that develops standards for computer networking and publishes RFCs; also, the network of computers that implements those standards
<b>DAA</b>	Data Authentication Algorithm, a NIST standard authentication code defined in FIPS PUB 113	<b>ISO</b>	International Standards Organization, an international standards body
<b>DEA</b>	Data Encryption Algorithm, the secret-key cryptosystem specified by DES	<b>MD2</b>	Message Digest Algorithm 2, a hash function developed by Ron Rivest that is defined in Internet RFC 1319
<b>DES</b>	Data Encryption Standard, a NIST standard defined in FIPS PUB 46-1 that specifies DEA	<b>MD5</b>	Message Digest Algorithm 5, another hash function developed by Ron Rivest and defined in Internet RFC 1321
<b>Diffie-Hellman</b>	A key-agreement algorithm invented by Whitfield Diffie and Martin Hellman	<b>MDC-2</b>	Manipulation Detection Code 2, the hash function specified in draft ANSI X9.31 part 2
<b>DSA</b>	Digital Signature Algorithm, the digital signature scheme specified by DSS	<b>NBS</b>	National Bureau of Standards; see NIST
<b>DSS</b>	Digital Signature Standard, a proposed NIST standard that specifies DSA	<b>NIST</b>	National Institute of Standards and Technology (formerly NBS), a US government agency that develops standards and publishes FIPS PUBs
<b>EDE</b>	Encrypt-decrypt-encrypt, a mode of DEA involving two keys and three DEA operations that is defined in ANSI X9.17	<b>OIW</b>	Open Systems Environment (formerly OSI) Implementors' Workshop, a group of developers that agrees on implementation issues such as algorithms
<b>EES</b>	Escrowed Encryption Standard, a proposed NIST standard that specifies Skipjack	<b>OSI</b>	Open Systems Interconnection, a standard networking model
<b>FIPS PUB</b>	Federal Information Processing Standard publication, one of a series of standards published by NIST		

are cited in the OIW implementors' agreements.<sup>36</sup> (As this article was going to press, I received a copy of Australian Standard AS28095.5.3, which specifies RSA.<sup>37</sup>)

**Digital signature schemes.** These schemes "sign" messages and verify the resulting signature with two different keys in such a way that it is difficult to sign without the signing key. Similar to public-key cryptosystems, the verification key can be published without compromising security, and is called the public key; the signing key is called the

private key.

Digital signature schemes provide integrity and origin authentication. Like public-key cryptosystems, they do not require that parties first agree on a secret key, and they are generally somewhat slower than, for instance, secret-key cryptosystems and cryptographic hash functions. They are often combined with hash functions to gain the benefits of both.

Public-key cryptosystems and digital signature schemes are

**Glossary (continued)**

<b>PEM</b>	Privacy-enhanced mail, a proposed Internet standard for encrypting and authenticating electronic mail; defined in Internet RFCs 1421-1424
<b>PKCS</b>	Public-key cryptography standards, informal standards developed by RSA Laboratories with representatives of Apple, Digital, Lotus, Microsoft, MIT, Northern Telecom, Novell, and Sun; available from RSA Laboratories or via electronic mail to pkcs@rsa.com
<b>RC4</b>	Rivest Cipher 4, a fast secret-key cryptosystem developed by Ron Rivest and proprietary to RSA Data Security
<b>RFC</b>	"Request for Comments," an Internet publication
<b>RSA</b>	Rivest-Shamir-Adleman algorithm, a public-key cryptosystem and digital signature scheme invented by Ron Rivest, Adi Shamir, and Len Adleman
<b>SC6</b>	Subcommittee 6 (Telecommunications and Information Exchange Between Systems), a joint subcommittee of ISO/IEC
<b>SC27/WG2</b>	Subcommittee 27 (Information Technology), Working Group 2 (Security Techniques), a joint working group of ISO/IEC
<b>SHA</b>	Secure Hash Algorithm, the hash function specified by SHS
<b>SHS</b>	Secure Hash Standard, a NIST standard defined in FIPS PUB 180 that specifies SHA
<b>SILS</b>	Secure Interoperable Local Area Network Security, an IEEE project; also called P802.10
<b>Skipjack</b>	The classified secret-key cryptosystem specified by EES
<b>SNMP</b>	Simple Network Management Protocol, an Internet standard defined in Internet RFC 1157
<b>Standards</b>	
<b>Australia</b>	An Australian standards body
<b>X9</b>	See ASC X9

closely related. In so-called reversible cryptography, signing in a digital signature scheme is the same as decryption in a public-key cryptosystem, while verification is the same as encryption. In irreversible cryptography, the relationships do not hold, although a given public/private-key pair may work in both a digital signature scheme and a public-key cryptosystem.

There is no primary standard digital signature scheme, but two main efforts are in progress. One involves RSA, which is reversible, and the other involves an irreversible algorithm

proposed by the US National Institute of Standards and Technology (NIST).

ISO/IEC 9796<sup>28</sup> almost creates a standard for RSA, but not quite. It defines a signature block format; RSA is in an informative (but nonstandard) annex. The block format prevents certain mathematical relationships among possible RSA signatures.<sup>29</sup> The draft ANSI X9.31 part 1,<sup>30</sup> which is expected to become a standard late this year, is based on ISO/IEC 9796 and specifies RSA. The intervender PKCS #1<sup>31</sup> gives alternate block formats for RSA signatures. ISO/IEC's joint working group SC27/WG2 is developing other digital signature standards.

NIST's proposed Digital Signature Standard (DSS),<sup>31</sup> which defines the Digital Signature Algorithm (DSA), has been the center of recent controversy.<sup>32</sup> DSA, an irreversible algorithm, is a variant of signature schemes due to Elgamal<sup>33</sup> and Schnorr.<sup>34</sup> It is intended to be combined with the Secure Hash Algorithm (SHA).<sup>35</sup> Mainly due to objections from industry, DSS has not yet been approved. The draft ANSI X9.30 part 1<sup>36</sup> specifies DSA.

**Key-agreement algorithms.** These algorithms manage keys through an exchange of messages derived from private values that are not shared. The result of the exchange is that parties agree on a secret key. It is difficult to determine the secret key from the exchanged messages without the private values from which they are derived. Key-agreement algorithms are sometimes called key exchange algorithms in the literature.

Key-agreement algorithms provide confidentiality and key management, and in some cases origin authentication. They do not require that parties first agree on a secret key. As with public-key cryptosystems, no primary standard key-agreement algorithm exists. Many consider an algorithm invented by Diffie and Hellman,<sup>21</sup> usually called Diffie-Hellman, the de facto standard here.

Efforts toward Diffie-Hellman standardization include the intervender PKCS #3<sup>37</sup> and the draft ANSI X9.30 part 4,<sup>38</sup> which is based on a variant of Diffie-Hellman having origin authentication. The cellular digital packet data (CDPD) specifications<sup>39</sup> adopt Diffie-Hellman key agreement. ISO/IEC's joint working group SC6 is developing standards for key agreement in the network and transport layers of the OSI reference model,<sup>39,40</sup> with Diffie-Hellman as a possible algorithm.

**Cryptographic hash functions.** These functions reduce a message of arbitrary length to a short code so that it is difficult to find a message with a given hash code, and in some cases also to find two messages with the same hash code. There is no key. Hash functions are also called message digests and modification detection codes in the literature.

A hash code is typically 128 or 160 bits long. Ideally, an attacker's only approach is trial and error, which amounts to  $2^{128}$  trials to find a message with a given hash code (for a 128-bit hash), and  $2^{160}$  trials to find two messages with the same hash code. (This is akin to the "birthday paradox": You need



### Encryption standards

365 people in a room to be likely to find one with a given birthday, but only 23 to be likely to find two with the same birthday.) Hash functions are generally quite fast. They provide message integrity to parties knowing a message's hash code. They are often combined with digital signature schemes, as noted earlier.

The Secure Hash Standard (SHS),<sup>35</sup> which defines SHA, is the primary standard. SHA produces a 160-bit hash from a message of arbitrary length; it is intended to be combined with DSA.<sup>31</sup> ANSI X9.30 part 2<sup>41</sup> specifies SHA.

Other hash algorithms suitable for standardization include MD2 and MD5, developed by Ron Rivest for RSA Data Security<sup>42,43</sup> and adopted by Internet privacy-enhanced mail,<sup>25</sup> and MDC-2, which is specified in draft ANSI X9.31 part 2.<sup>44</sup> SC27/WG2 is also developing standards for hash functions.

**Authentication codes.** These codes reduce a message of arbitrary length to a short code under a secret key so that it is difficult, without the key, to compute the authentication code, or to find a new message with a given authentication code. Authentication codes provide message integrity and origin authentication to parties who have previously agreed on a secret key. The message itself need not be encrypted.

An authentication code is typically 32 or 64 bits long, and the keys are 56 bits long. Ideally, an attacker's only approach is trial and error on the keys; arbitrary message modifications have some probability of success, but the attacker cannot check for success without the help of the real user. Authentication codes, like hash functions, are generally quite fast.

The primary standard is FIPS PUB 113,<sup>45</sup> which defines the Data Authentication Algorithm. The algorithm is a variant of DEA; it produces a 32-bit authentication code from a message of arbitrary length and a 56-bit key. ANSI X9.9<sup>46</sup> and Australian standard AS2805.4<sup>47</sup> specify DAA.

### **Applications**

The applications standards described next combine families of algorithms, and sometimes specify particular algorithms, to solve confidentiality, integrity, origin authentication, and key management problems. Although many of the standards specify much more than just cryptography, encryption plays an important role.

Ideally, an algorithm should work in many applications, and many algorithms should work in a given application. The design of applications and algorithms is in this sense "orthogonal," and the designers have generally done a good job at providing orthogonality.

Do not confuse these applications with the applications layer of the OSI reference model; some may well run at that layer, and others at lower layers.

**Secure electronic mail.** Six years in development and now a proposed standard, Internet privacy-enhanced mail (PEM) combines secret-key cryptosystems, public-key cryptosystems, hash functions, and digital signature schemes

to provide security for electronic mail.<sup>48</sup> It is a text-based protocol compatible with most electronic-mail systems. PEM supports public-key and secret-key techniques; the former involves X.509 certificates.<sup>49</sup> Currently, PEM has adopted RSA, DEA, MD2, and MD5 algorithms,<sup>25</sup> but the protocols are flexible and other suites of algorithms are likely to be added.

Mail is not the only application of PEM, of course, although it is a primary one. The same protocol that adds encryption or authentication to a mail message can enhance any digital document, such as a contract; the document need not be mailed to someone.

The intervender PKCS #7<sup>50</sup> is a binary extension of PEM; it offers the same services, but works with binary data and allows one to sign attributes such as the time of day along with the underlying message. Certain modes of PKCS #7 are cryptographically compatible with PEM, in the sense that messages can be translated between the two protocols without any cryptographic operations. PKCS #7 does not specify a particular algorithm.

Another approach to electronic-mail security is found in X.400 message-handling systems,<sup>51</sup> which solve the basic problems of confidentiality, authentication, and key management. X.400 also provides special encryption-based services such as proof of submission and proof of delivery. (X.411 supplies the details.<sup>52</sup>) X.400, like most international standards, does not specify particular algorithms. It supports both public-key and secret-key techniques. ISO 10021-1<sup>53</sup> is technically aligned with X.400.

X.435,<sup>54</sup> a standard for electronic data interchange over X.400, builds on X.411's services, defining related services such as signed receipts.

**Secure communications.** These standards focus on the security of local-area networks and wireless links. IEEE's P802.10 project, Secure Interoperable LAN (local area network) Security (SILS), addresses privacy and authentication of data at the data link layer. Devices following the protocol encrypt data link frames as they pass through the network; the protocol is transparent to higher layers. A proposed draft<sup>55</sup> specifies Diffie-Hellman key agreement. The CDPD specifications<sup>20</sup> define an encryption protocol for wireless links based on Diffie-Hellman key agreement and RC4.

IEEE project P802.11, focusing on wireless links, has just started.

**Directory authentication and network management.** X.509 directory authentication<sup>49</sup> applies public-key and secret-key techniques to the problem of determining the identity of a user attempting to access an X.500 global directory.<sup>56</sup> "Weak" authentication identifies a user by a password, while "strong" authentication involves digital signatures. The authentication protocols can also ensure that messages to and from the directory are not modified in transit.

X.509 standardizes on no particular algorithm, although RSA is in an informative annex. Two additional contributions

of X.509 are certificates, which bind a public key to a user's name with a digital signature, and certificate-revocation lists, which break the binding. These elements have found their way into other applications such as privacy-enhanced mail and the X9.30 and X9.31 drafts. Although directories are just emerging, users' names in the related applications are designed in anticipation of a future directory entry. ISO 9594-8<sup>57</sup> is technically aligned with X.509.

In a proposed security standard for the Internet's Simple Network Management Protocol (SNMP),<sup>58</sup> parties identify each other with a secret shared key.<sup>59</sup> Network management requests are hashed together with the secret key under MD5 to produce an authentication code. Encryption with DEA is also an option.

SC27/WG2 is developing authentication protocols involving public-key and secret-key techniques.

**Banking.** The primary key management standard for the banking industry is ANSI X9.17. It is based entirely on DEA and related algorithms, including the EDE mode of DEA. To date, X9's standards have all involved secret-key techniques; work on public-key techniques is in progress in X9.30 and X9.31. Other banking standards efforts include

- draft Australian standard AS2805.6.5.3,<sup>60</sup> which specifies RSA;
- CFONB ETEBAC-5,<sup>61</sup> a French banking standard that specifies RSA and DEA; and
- ISO CD 11666, a draft standard for banking key management that specifies RSA.<sup>62,63</sup> Whether it will be approved is unclear, as its architectural features have been criticized.<sup>64</sup>

**Escrowed encryption.** A likely candidate to surpass even the DSS controversy is the proposed Escrowed Encryption Standard (EES),<sup>17</sup> part of the US government's Capstone project for encryption standards. It implements an April 1993 presidential order that certain encryption devices provide entry points for legitimate law-enforcement wiretaps. The government's Clipper chips are the first examples of such devices.<sup>65</sup>

EES is based on the Skipjack algorithm and involves a classified law-enforcement access field (LEAF). Each hardware device complying with EES (software is not allowed) has a secret key; the key is split at the factory and "escrowed" with (that is, put into the custody of, as with money or deeds) two government agencies. Under court order, the agencies reconstruct the key. With the secret key and LEAF, authorized officials can decrypt messages encrypted by the device. Neither escrow agency can decrypt messages by itself.


What is controversial about EES appears not so much to be government wiretapping, which has always been controversial, but the issues of algorithm secrecy, hardware-only implementation, and potential security risks in the manufac-

turing and key escrow processes. The panel that reviewed the Skipjack algorithm is also evaluating the manufacturing and key escrow processes.

CRYPTOGRAPHY IS FINDING BROAD APPLICATION in the computer world. There is much common ground in the underlying algorithms. Interestingly, solutions to the confidentiality problem—encryption in the pure sense—seem to be the hardest to standardize. Much more activity focuses on peripheral cryptographic problems such as authentication and key management, as well as algorithm-independent standards.

As evidenced by the parallel X9.30 and X9.31 efforts, the controversy over DSS has brought about parallel standards, one involving the reversible model (for example, RSA). Here, signing is the same as encryption, and verification is the same as decryption. The other standard involves the irreversible model (for example, DSA) without such relationships. Reversibility is considered by some to open the door to confidentiality of unlimited security, a problematic feature for law enforcement and national security concerns. Others see dual standardization to be problematic for industry concerns.

Since NIST may have reaffirmed DES for the last time, what comes next? The Internet's PEM working group has been looking at new encryption algorithms, among them the so-called triple-DES with three DEA operations, of which X9.17's EDE is one example. Whether the factor-of-three slowdown in performance is too much remains to be seen, but in light of the secrecy around the Skipjack algorithm and the few published alternatives, most likely triple-DES will become a standard encryption algorithm in some corner of the standards world. RC4 may play a role as well.

While all of this is sorting itself out, a new IEEE project, sponsored by the Computer Society's Microprocessor and Microcomputer Standards Committee, aims to complete the family of public-key standards. These standards will be based on the RSA and Diffie-Hellman algorithms, covering key management, encryption, authentication, key generation, and hardware support. The IEEE authorized P1363, "RSA, Diffie-Hellman, and related public-key techniques" this June, and an initial meeting is being planned as of this writing. 

## Acknowledgments

I am grateful to Richard Ankney for sharing his standards expertise.

## References

1. W. Diffie, "The First Ten Years of Public-Key Cryptography," *Proc. IEEE*, 1988, pp. 560-577.
2. G.J. Simmons, ed., *Contemporary Cryptology: The Science of*

Encryption standards

- Information Integrity, IEEE, New York, 1992.
3. P. Fahn, *Answers to Frequently Asked Questions About Today's Cryptography, Version 2.0*, RSA Laboratories, Redwood City, Calif., Sept. 1993.
  4. A. Patel, "Emerging Network Security Standards in an OSI Environment," *Computer Standards & Interfaces*, 1989/1990, pp. 239-247.
  5. *Recommendation X.800: Security Architecture for Open Systems Interconnection for CCITT Applications*, CCITT, Geneva, 1991.
  6. *ISO/IEC DIS 11586: Generic Upper Layers Security*, ISO/IEC, Geneva, 1993.
  7. Accredited Standards Committee X9, *Working Draft: American National Standard X9.41-1993: Security Services Management for the Financial Services Industry*, American Bankers Assoc., Washington, D.C., Aug. 1993.
  8. *DoD 5200.28-STD: Department of Defense (DoD) Trusted Computer System Evaluation Criteria (TCSEC)*, US Department of Defense, Washington, D.C., 1985.
  9. *FIPS Publication 46-1: Data Encryption Standard*, NIST, Washington, D.C., Jan. 22, 1988; originally issued by the National Bureau of Standards.
  10. Accredited Standards Committee X3, *ANSI X3.92: Data Encryption Algorithm (DEA)*, ANSI, New York, 1981.
  11. *Australian Standard 2805.5 1985: Electronics Funds Transfer—Requirements for Interfaces: Part 5—Data Encryption Algorithm*, Standards Assoc. of Australia, North Sydney, NSW, 1985.
  12. E. Biham and A. Shamir, "Differential Cryptanalysis of the Full 16-Round DES," *Proc. Crypto 92, Advances in Cryptology*, Springer-Verlag, New York, 1993, to appear.
  13. D. Coppersmith, "The Data Encryption Standard (DES) and Its Strength Against Attacks," tech. report RC 18613 (81421), IBM Research Div., Yorktown Heights, N.Y., Dec. 1992.
  14. *FIPS Publication 81: DES Modes of Operation*, NIST, Dec. 2, 1980.
  15. Accredited Standards Committee X9, *American National Standard X9.17: Financial Institution Key Management (Wholesale)*, ANSI, 1985.
  16. *PKCS #5: Password-Based Encryption Standard, Version 1.4*, RSA Data Security, Inc., Redwood City, Calif., June 1991.
  17. NIST, "A Proposed Federal Information Processing Standard for an Escrowed Encryption Standard (EES)," *Federal Register*, Vol. 58, No. 145, July 30, 1993.
  18. E.F. Brickell et al., "Skipjack Review, Interim Report: The Skipjack Algorithm," July 28, 1993; contact author for copies.
  19. R.L. Rivest, *The RC4 Encryption Algorithm*, RSA Data Security, Inc., Mar. 12, 1992.
  20. Ameritech Mobile Communications et al., *Cellular Digital Packet Data System Specifications: Part 406: Airlink Security*, CDPD Industry Input Coordinator, Costa Mesa, Calif., July 1993.
  21. W. Diffie and M.E. Hellman, "New Directions in Cryptography," *IEEE Trans. Information Theory*, Vol. IT-22, 1976, pp. 644-654.
  22. R.L. Rivest, A. Shamir, and L. Adleman, "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems," *Comm. ACM*, Vol. 21, No. 2, Feb. 1978, pp. 120-126.
  23. *PKCS #1: RSA Encryption Standard, Version 1.4*, RSA Data Security, Inc., June 1991.
  24. Accredited Standards Committee X9, *Working Draft: American National Standard X9.31-1993: Public Key Cryptography Using Reversible Algorithms for the Financial Services Industry: Part 4: Management of Symmetric Algorithm Keys Using RSA*, Am. Bankers Assoc., June 4, 1993.
  25. D. Balenson, *RFC 1423: Privacy Enhancement for Internet Electronic Mail: Part III—Algorithms, Modes, and Identifiers*, Trusted Information Systems, Inc., Glenwood, Md., Feb. 1993.
  26. *Special Publication 500-183: Stable Implementation Agreements for Open Systems Interconnection Protocols: Part 12—OS Security*, NIST, June 1992.
  27. *Australian Standard 2805.5.3—Electronic Data Transfer—Requirements for Interfaces: Part 5.3: Data Encipherment Algorithm 2*, Standards Assoc. of Australia, 1992.
  28. *International Standard 9796: Information Technology, Security Techniques: Digital Signature Scheme Giving Message Recovery*, ISO/IEC, 1991.
  29. L.C. Guillou et al., "Precautions Taken Against Various Potential Attacks in ISO/IEC DIS 9796," *Proc. Eurocrypt 90, Advances in Cryptology*, I.B. Damgard, ed., Springer-Verlag, 1991, pp. 465-473.
  30. Accredited Standards Committee X9, *Working Draft: American National Standard X9.31-1992: Public Key Cryptography Using Reversible Algorithms for the Financial Services Industry: Part 1: The RSA Signature Algorithm*, Am. Bankers Assoc., Mar. 7, 1993.
  31. *Publication XX: Announcement and Specifications for a Digital Signature Standard (DSS)*, NIST, Aug. 19, 1992.
  32. NIST, "The Digital Signature Standard, Proposal and Discussion," *Comm. ACM*, Vol. 35, No. 7, July 1992, pp. 36-54.
  33. T. Elgamal, "A Public-Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms," *IEEE Trans. Information Theory*, Vol. IT-31, 1985, pp. 469-472.
  34. C.P. Schnorr, "Efficient Identification and Signatures for Smart Cards," *Proc. Crypto 89, Advances in Cryptology*, Springer-Verlag, 1990, pp. 239-251.
  35. *FIPS Publication 180: Secure Hash Standard (SHS)*, NIST, May 11, 1993.
  36. Accredited Standards Committee X9, *Working Draft: American National Standard X9.30-199X: Public Key Cryptography Using Irreversible Algorithms for the Financial Services Industry: Part 1: The Digital Signature Algorithm (DSA)*, Am. Bankers Assoc., June 18, 1993.
  37. *PKCS #3: Diffie-Hellman Key Agreement Standard, Version 1.3*, RSA Data Security, Inc., June 1991.
  38. Accredited Standards Committee X9, *Working Draft: American National Standard X9.30-1993: Public Key Cryptography Using Irreversible Algorithms for the Financial Services Industry: Part 4: Management of Symmetric Algorithm Keys Using Diffie-Hellman*, Am. Bankers Assoc., June 4, 1993.
  39. *ISO/IEC JTC 1/SC 6, N2559: Draft Network Layer Security Protocol*, ISO/IEC, Sept. 1990.



40. ISO/IEC JTC1/SC6, N6285: *Draft Transport Layer Security Protocol*, ISO/IEC, Nov. 1990.
41. Accredited Standards Committee X9, *Working Draft: American National Standard X9.30-1993: Public Key Cryptography Using Irreversible Algorithms for the Financial Services Industry: Part 2: The Secure Hash Algorithm (SHA)*, Am. Bankers Assoc., 1993.
42. B. Kaliski, *RFC 1319: The MD2 Message-Digest Algorithm*, RSA Data Security, Inc., Apr. 1992.
43. R.L. Rivest, *RFC 1321: The MD5 Message-Digest Algorithm*, MIT Laboratory for Computer Science, Cambridge, Mass., and RSA Data Security, Inc., Apr. 1992.
44. Accredited Standards Committee X9, *Working Draft: American National Standard X9.31-1992: Public Key Cryptography Using Reversible Algorithms for the Financial Services Industry: Part 2: The MDC-2 Hash Algorithm*, Am. Bankers Assoc., June 4, 1993.
45. *FIPS Publication 113: Computer Data Authentication*, NIST, May 30, 1985.
46. Accredited Standards Committee X9, *American National Standard X9.9: Financial Institution Message Authentication*, ANSI, 1982.
47. *Australian Standard 2805.4 1985: Electronics Funds Transfer—Requirements for Interfaces: Part 4—Message Authentication*, Standards Assoc. of Australia, 1985.
48. J. Linn, *RFC 1421: Privacy Enhancement for Internet Electronic Mail: Part I—Message Encipherment and Authentication Procedures*, Internet Activities Board, Feb. 1993.
49. *Recommendation X.509: The Directory—Authentication Framework*, CCITT, 1988.
50. *PKCS #7: Cryptographic Message Syntax Standard, Version 1.4*, RSA Data Security, Inc., June 1991.
51. *Recommendation X.400: Message Handling System and Service Overview*, CCITT, 1988.
52. *Recommendation X.411: Message Handling Systems: Message Transfer System: Abstract Service Definition and Procedures*, CCITT, 1988.
53. *International Standard 10021-1: Information Processing Systems, Text Communication, MOTIS, Message Handling: System and Service Overview*, ISO/IEC, 1988.
54. *Recommendation X.435: Message Handling Systems: EDI Messaging System*, CCITT, 1991.
55. *IEEE Draft Standard for Interoperable Local Area Network (LAN) Security: Part C—Key Management Proposal*, IEEE, Apr. 10, 1992.
56. *Recommendation X.500: The Directory—Overview of Concepts, Models and Services*, CCITT, 1988.
57. *International Std. 9594-8: Information Processing Systems—Open Systems Interconnection—The Directory—Authentication Framework*, ISO/IEC, 1988.
58. J. Case et al., *RFC 1157: The Simple Network Management Protocol*, SNMP Research, Knoxville, Tenn., May 1990.
59. J. Galvin, K. McCloghrie, and J. Davin, *RFC 1352: SNMP Security Protocols*, Trusted Information Systems, Inc., July 1992.
60. *Draft Australian Standard: Electronics Funds Transfer—Requirements for Interfaces: Part 6.5.3—Key Management—Terminal Initialisation—Asymmetrical*, Standards Assoc. of Australia, May 1990.
61. *Echanges Télématiques entre les Banques et leurs Clients, Standard ETEBAC 5, v1.1*, Comité Français d'Organisation et de Normalisation Bancaire, Paris, 1989.
62. *ISO CD 11166-1, Banking—Key Management by Means of Asymmetric Algorithms—Part 1: Principles, Procedures and Formats*, ISO TC68/SC2/N346, 1992.
63. *ISO CD 11166-2, Banking—Key Management by Means of Asymmetric Algorithms—Part 2: Approved Algorithms Using the RSA Cryptosystem*, ISO TC68/SC2/N304, 1992.
64. *ISO/IEC JTC1/SC27/WG2, N205: Criticism of ISO CD 11666 Banking—Key Management by Means of Asymmetric Algorithms*, ISO/IEC, Apr. 1993.
65. D.E. Denning, "The Clipper Encryption System," *Am. Scientist*, Vol. 81, No. 4, July-Aug. 1993, pp. 319-323.



**Burt Kaliski** is chief scientist of the RSA Laboratories division of RSA Data Security. His research interests include cryptography and fast arithmetic.

Kaliski received the BS, MS, and PhD degrees in computer science from the Massachusetts Institute of Technology, Cambridge. He is a member of the IEEE Computer Society, the International Association for Cryptologic Research, Sigma Xi, and Tau Beta Pi. He served as general chair of Crypto 91, and he is interim chair of IEEE P1363.

Direct questions concerning this article to the author at RSA Laboratories, 100 Marine Parkway, Redwood City, CA 94065; or burt@rsa.com.

#### Reader Interest Survey

Indicate your interest in this article by circling the appropriate information on the Reader Service Card.

Low 171                      Medium 172                      High 173